Amendments to the Claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): An apparatus for determining a longest matching prefix, the apparatus comprising:

an external memory for storing one or more non-first-level tiny trees; and

an applications specific integrated circuit (ASIC) including a lookup engine and internal memory for storing a set of first-level tiny trees, the ASIC configured to perform operations, said operations including:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value on T disjoint ranges defined by endpoints of prefixes represented by said first-level tiny trees and said one or more non-first-level tiny trees, with the endpoint of a fully expanded prefix of said represented prefixes being the value of the fully expanded prefix itself and the endpoints of a non-fully expand prefix being the non-fully expand prefix fully expanded with zeros and the non-fully expand prefix fully expanded with ones, with each of said T disjoint ranges including substantially the same number of said endpoints;

retrieving a first-level root node of said particular first-level tiny tree from said internal memory, the first-level root node including a first-level plurality of keys defining a plurality of tiny intervals, wherein a tiny interval is a range defined by two consecutive keys;

traversing said particular first-level tiny tree stored in said internal memory to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys defining a plurality of tiny intervals, wherein the first-level tiny tree and the next-level tiny tree are independent trees;

retrieving a root node of the next-level tiny tree stored in the external memory, the root node including a plurality of keys, defining a plurality of tiny intervals, to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree stored in the external memory to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys defining a plurality of tiny intervals; and

identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

Claim 2 (original): The apparatus of claim 1, comprising an associative memory, wherein said determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value includes performing a lookup operation on the associative memory based on the lookup value.

Claim 3 (currently amended): A method for identifying a longest matching prefix, the method comprising:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value on T disjoint ranges defined by endpoints of prefixes represented by said first-level tiny trees and said one or more non-first-level tiny trees, with the endpoint of a fully expanded prefix of said represented prefixes being the value of the fully expanded prefix itself and the endpoints of a non-fully expand prefix being the non-fully expand prefix fully expanded with zeros and the non-fully expand prefix fully expanded with ones, with each of said T disjoint ranges including substantially the same number of said endpoints;

retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys defining a plurality of tiny intervals, wherein a tiny interval is a range defined by two consecutive keys;

traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys defining a plurality of tiny intervals, wherein the first-level tiny tree and the next-level tiny tree are independent trees;

retrieving a root node of the next-level tiny tree, the root node including a plurality of keys, defining a plurality of tiny intervals, to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys defining a plurality of tiny intervals; and

identifying and storing as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

Claim 4 (previously presented): The method of claim 3, wherein the first-level tiny tree is associated with no back values.

Claim 5 (currently amended): A tangible computer-readable medium storing therein computer-executable instructions for performing steps for identifying a longest matching prefix, said steps comprising:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value on T disjoint ranges defined by endpoints of prefixes represented by said first-level tiny trees and said one or more non-first-level tiny trees, with the endpoint of a fully expanded prefix of said represented prefixes being the value of the fully expanded prefix itself and the endpoints of a non-fully expand prefix being the non-fully expand prefix fully expanded with zeros and the non-fully expand prefix fully expanded with ones, with each of said T disjoint ranges including substantially the same number of said endpoints;

retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys defining a plurality of tiny intervals, wherein a tiny interval is a range defined by two consecutive keys;

traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys defining a plurality of tiny intervals, wherein the first-level tiny tree and the next-level tiny tree are independent trees;

retrieving a root node of the next-level tiny tree, the root node including a plurality of keys, defining a plurality of tiny intervals, to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys defining a plurality of tiny intervals; and

identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

Claim 6 (currently amended): An apparatus for identifying a longest matching prefix, the method comprising:

means for determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value on T disjoint ranges defined by endpoints of prefixes represented by said first-level tiny trees and said one or more non-first-level tiny trees, with the endpoint of a fully expanded prefix of said represented prefixes being the value of the fully expanded prefix itself and the endpoints of a non-fully expand prefix being the non-fully expand prefix fully expanded with zeros and the non-fully expand prefix fully expanded with ones, with each of said T disjoint ranges including substantially the same number of said endpoints;

means for retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys defining a plurality of tiny intervals, wherein a tiny interval is a range defined by two consecutive keys;

means for traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys defining a plurality of tiny intervals, wherein the first-level tiny tree and the next-level tiny tree are independent trees;

means for retrieving a root node of the next-level tiny tree, the root node including a plurality of keys, defining a plurality of tiny intervals, to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

means for traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys defining a plurality of tiny intervals; and

means for identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

Claim 7-23 (cancelled)

Claim 24 (currently amended): An apparatus comprising one or more processors and a memory, wherein the memory stores one or more instructions that, when executed by the one or more processors, perform the operations of:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value on T disjoint ranges defined by endpoints of prefixes represented by said first-level tiny trees and said one or more non-first-level tiny trees, with the endpoint of a fully expanded prefix of said represented prefixes being the value of the fully expanded prefix itself and the endpoints of a non-fully expand prefix being the non-fully expand prefix fully expanded with zeros and the non-fully expand prefix fully expanded with ones, with each of said T disjoint ranges including substantially the same number of said endpoints;

retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys defining a plurality of tiny intervals, wherein a tiny interval is a range defined by two consecutive keys;

traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys defining a plurality of tiny intervals, wherein the first-level tiny tree and the next-level tiny tree are independent trees;

retrieving a root node of the next-level tiny tree, the root node including a plurality of keys, defining a plurality of tiny intervals, to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys defining a plurality of tiny intervals; and

identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

Claim 25 (previously presented): The apparatus of claim 24, wherein the first-level tiny tree is associated with no back values.